UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/621,908	07/17/2003	Sang Hoo Dhong	AUS920030428US1	2141

50170 7590 07/12/2006

IBM CORP. (WIP)
c/o WALDER INTELLECTUAL PROPERTY LAW, P.C.
P.O. BOX 832745
RICHARDSON, TX 75083

EXAMINER	
GEIB, BENJAMIN P	

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 07/12/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 10/621,908	Applicant(s) DHONG ET AL.	
	Examiner Benjamin P. Geib	Art Unit 2181	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 April 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-3,8,11-14 and 19-32* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *19-28 and 32* is/are allowed.

6)☒ Claim(s) *1-3,8,11-14 and 29-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 July 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Fritz Fleming

FRITZ FLEMING
Supervisory PRIMARY EXAMINER 6/26/2006
GROUP 2100
AU 2181

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-3, 8, 11-14, and 19-32 have been examined.

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as received on 04/19/2006.

Allowable Subject Matter

3. Claims 19-28, and 32 are allowed.

Claim Rejections - 35 USC § 103

4. Claims 1-3 and 29-31 are rejected 35 U.S.C. 103(a) as being unpatentable over Chehrazi et al., U.S. Patent No. 6,282,556, (Herein referred to as Chehrazi) in view of Peleg et al., U.S. Patent No. 6,070,237, (Herein referred to as Peleg) and further in view of "AltiVec Technology Programming Environments Manual" (Herein referred to as AltiVec).

5. Referring to claim 1, Chehrazi has taught a byte execution unit, comprising:

logic coupled to receive byte instruction information, to receive a first operand from a first source register, to receive a second operand from a second source register, and configured to perform an operation specified by the byte instruction information upon at least one of the first operand or the second operand, thereby producing a result in a destination register *(column 7, lines 21-51)*, wherein the byte instruction information specifies either a count ones in bytes operation, an average bytes operation *(See Table*

I, instruction AVG), an absolute differences of bytes operation *(column 3, lines 49-54)*, or a sum bytes into halfwords operation.

wherein responsive to an average bytes operation, the byte execution unit averages each byte of the first operand with a corresponding byte of the second operand and stores each result in a corresponding byte of the destination register *(column 13, lines 12-20)*;

wherein responsive to an absolute differences of bytes operation, the byte execution unit subtracts each byte of the first operand from a corresponding byte of the second operand to form an intermediate result, determines an absolute value of each intermediate result to form a final result, and stores each final result in a corresponding byte of the destination register *(column 3, lines 49-54, column 9, lines 6-15)*; and

Chehrazi has not taught that the byte instruction information can specify a count ones in bytes operation and, wherein responsive to a count ones in bytes operation, the byte execution unit counts a number of logical one bits in each byte of at least the first operand and stores each result in a corresponding byte of the destination register.

Peleg has taught byte instruction information specifying a count ones in bytes operation *(Peleg; population count operation)* and wherein responsive to a count ones in bytes operation, a byte execution unit counts a number of logical one bits in each byte of at least a first operand and stores each result in a corresponding byte of a destination register *(Peleg; column 10, line 47 – column 11, line 17)*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi to include the count ones in bytes operation and required circuitry as taught by Peleg.

The suggestion/motivation for doing so would have been that the performance of algorithms that require the totaling of the number of bits set is improved *(Peleg; column 4, lines 13-16)*.

Chehrazi and Peleg have not taught that the byte instruction information can specify a sum bytes into halfwords operation and, wherein responsive to a sum bytes into halfwords operation, the byte execution unit sums a number of corresponding one-byte portions of the first operand or the second operand and stores each result in a corresponding halfword of the destination register.

AltiVec has taught byte instruction information specifying a sum bytes into halfwords operation *[AltiVec; Vector Sum Across Partial (1/4) Unsigned Byte Saturate (vsum4ubs) operation; page 6-170]* and, wherein responsive to a sum bytes into halfwords operation, a byte execution unit sums a number of corresponding one-byte portions of a first operand or a second operand and stores each result in a corresponding halfword of a destination register *[AltiVec; Responsive to a vsum4ubs operation, 4 one-byte portions of the vA operand are summed and the results are stored in a corresponding 32-bit halfword (i.e. half of a 64-bit word); page 6-170]*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi and Peleg to include the sum bytes into halfwords operation and required circuitry as taught by AltiVec.

The suggestion/motivation for doing so would have been that the performance of many common vector algorithms, such as dot-product algorithms, is improved.

6. Referring to claim 2, Chehrazi has taught the byte execution unit as recited in claim 1, wherein each the first operand and the second operand comprises a plurality of bits, and wherein the bits of each of the first operand and the second operand are grouped to form a plurality of corresponding 8-bit bytes *(column 7, lines 23-27; See Fig. 2, component 210)*.

7. Referring to claim 3, Chehrazi has taught the byte execution unit as recited in claim 2, wherein each of the first operand and the second operand comprises 128 bits, and wherein the bits of each of the first operand and the second operand are grouped to form 16 corresponding bytes *(column 7, lines 23-27; See Fig. 2, component 210)*.

8. Referring to claim 29, Chehrazi has taught a data processing system, comprising:

a memory system *(Fig. 1, component 102; column 5, lines 46-54)* comprising a byte instruction, wherein the byte instruction specifies either a count ones in bytes operation, an average bytes operation *(See Table I, instruction AVG)*, an absolute differences of bytes operation *(column 3, lines 49-54)*, or a sum bytes into halfwords operation; and

a processor *(multimedia coprocessor; Fig. 1, component 108)* coupled to the memory system and configured to fetch and execute instructions from the memory system *(column 7, lines 21-40)*, wherein the processor comprises:

a byte execution unit coupled to receive byte instruction information, to receive a first operand from a first source register, and to receive a second operand from a second source register, and configured to perform an operation specified by the byte instruction information upon at least one of the first operand or the second operand, thereby producing a result in a destination register *(column 7, lines 21-51)*.

wherein responsive to an average bytes operation, the byte execution unit averages each byte of the first operand with a corresponding byte of the second operand and stores each result in a corresponding byte of the destination register *(column 13, lines 12-20)*;

wherein responsive to an absolute differences of bytes operation, the byte execution unit subtracts each byte of the first operand from a corresponding byte of the second operand to form an intermediate result, determines an absolute value of each intermediate result to form a final result, and stores each final result in a corresponding byte of the destination register *(column 3, lines 49-54, column 9, lines 6-15)*

Chehrazi has not taught that the byte instruction information can specify a count ones in bytes operation and, wherein responsive to a count ones in bytes operation, the

byte execution unit counts a number of logical one bits in each byte of at least the first operand and stores each result in a corresponding byte of the destination register.

Peleg has taught byte instruction information specifying a count ones in bytes operation *(Peleg; population count operation)* and wherein responsive to a count ones in bytes operation, a byte execution unit counts a number of logical one bits in each byte of at least a first operand and stores each result in a corresponding byte of a destination register *(Peleg; column 10, line 47 – column 11, line 17)*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi to include the count ones in bytes operation as taught by Peleg.

The suggestion/motivation for doing so would have been that the performance of algorithms that require the totaling of the number of bits set is improved *(Peleg; column 4, lines 13-16)*.

Chehrazi and Peleg have not taught that the byte instruction information can specify a sum bytes into halfwords operation and, wherein responsive to a sum bytes into halfwords operation, the byte execution unit sums a number of corresponding one-byte portions of the first operand or the second operand and stores each result in a corresponding halfword of the destination register.

AltiVec has taught byte instruction information specifying a sum bytes into halfwords operation *[AltiVec; Vector Sum Across Partial (1/4) Unsigned Byte Saturate (vsum4ubs) operation; page 6-170]* and, wherein responsive to a sum bytes into

halfwords operation, a byte execution unit sums a number of corresponding one-byte portions of a first operand or a second operand and stores each result in a corresponding halfword of a destination register *[AltiVec; Responsive to a vsum4ubs operation, 4 one-byte portions of the vA operand are summed and the results are stored in a corresponding 32-bit halfword (i.e. half of a 64-bit word); page 6-170]*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi and Peleg to include the sum bytes into halfwords operation as taught by AltiVec.

The suggestion/motivation for doing so would have been that the performance of many common vector algorithms, such as dot-product algorithms, is improved.

9. Referring to claim 30, given the similarities between claim 2 and claim 30 the arguments as stated for the rejection of claim 2 also apply to claim 30.

10. Referring to claim 31, given the similarities between claim 3 and claim 31 the arguments as stated for the rejection of claim 3 also apply to claim 31.

11. Claims 8 and 11-13 are rejected 35 U.S.C. 103(a) as being unpatentable over Chehrazi in view of Oberman et al., "AMD 3DNow! Technology: Architecture and Implementations", (Herein referred to as Oberman) in view of Peleg and further in view of AltiVec.

12. Referring to claim 8, Chehrazi has taught a byte execution unit, comprising:

pre-processing logic *(logic unit; Fig. 4, component 305)* coupled to receive a plurality of operands and configured to perform an operation upon the operands dependent upon an operation specified by a byte instruction, thereby producing an intermediate result *(See Fig. 4; column 8, lines 10-17)*;

adder logic *(carry propagate adder; Fig. 4, component 340)* coupled to receive the intermediate result and configured to perform an addition operation upon the intermediate result, thereby producing a sum *(See Fig. 4; column 8, lines 23-31)*;

post-processing logic *(saturation unit; Fig. 4, component 342)* coupled to receive the sum and configured to perform an operation *(saturation)* upon the sum dependent upon the operation specified by a byte instruction, thereby producing a result *(column 9, lines 45-67)*; and

a control unit coupled to the pre-processing logic, the adder logic, and the post-processing logic *[control signals are sent to the pre-processing logic, the adder logic, and the post-processing logic (See Fig. 4). There is inherently a controller that sends these control signals]*

wherein responsive to an average bytes operation, the control unit sets control signals to configure the pre-processing logic, the adder logic, and the post-processing logic to average each byte of the first operand with a corresponding byte of the second operand and to store each result in a corresponding byte of the destination register *(column 13, lines 12-20)*;

wherein responsive to an absolute differences of bytes operation, the control unit sets control signals to configure the pre-processing logic, the adder logic, and the post-

processing logic to subtract each byte of the first operand from a corresponding byte of the second operand to form an intermediate result, to determine an absolute value of each intermediate result to form a final result, and to store each final result in a corresponding byte of the destination register *(column 3, lines 49-54, column 9, lines 6-15)*;

Chehrazi has not explicitly taught that the adder logic produces a sum+1 and that the post-processing logic receives the sum+1 and performs an operation on it.

Oberman teaches adder logic *(Oberman; compound adder)* that produces a sum+1 output and post-processing logic *(Oberman; selection logic)* that receives the sum+1 and performs an operation on it *(Oberman; Last paragraph in first column on page 43)*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the adder logic and post-processing logic of Chehrazi to produce a sum+1 and receive the sum+1 and perform an operation on it, respectively, as taught by Oberman.

The suggestion/motivation for doing so would have been that the time required for rounding is reduced *(Oberman; Last paragraph in first column on page 43)*.

Chehrazi and Oberman have not taught that the byte instruction information can specify a count ones in bytes operation and, wherein responsive to a count ones in bytes operation, the control unit sets control signals to configure the pre-processing

logic, the adder logic, and the post-processing logic to count a number of logical one bits in each byte of at least the first operand and to store each result in a corresponding byte of the destination register.

Peleg has taught byte instruction information specifying a count ones in bytes operation *(Peleg; population count operation)* and wherein responsive to a count ones in bytes operation, a byte execution unit is configured to count a number of logical one bits in each byte of at least a first operand and to store each result in a corresponding byte of a destination register *(Peleg; column 10, line 47 – column 11, line 17)*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi and Oberman to include the count ones in bytes operation and required circuitry as taught by Peleg.

The suggestion/motivation for doing so would have been that the performance of algorithms that require the totaling of the number of bits set is improved *(Peleg; column 4, lines 13-16)*.

Chehrazi, Oberman, and Peleg have not taught that the byte instruction information can specify a sum bytes into halfwords operation and, wherein responsive to a sum bytes into halfwords operation, the control unit sets control signals to configure the pre-processing logic, the adder logic, and the post-processing logic to sum a number of corresponding one-byte portions of the first operand or the second operand and to store each result in a corresponding halfword of the destination register.

AltiVec has taught byte instruction information specifying a sum bytes into halfwords operation *[AltiVec; Vector Sum Across Partial (1/4) Unsigned Byte Saturate (vsum4ubs) operation; page 6-170]* and, wherein responsive to a sum bytes into halfwords operation, a byte execution unit is configured to sum a number of corresponding one-byte portions of a first operand or a second operand and to store each result in a corresponding halfword of a destination register *[AltiVec; Responsive to a vsum4ubs operation, 4 one-byte portions of the vA operand are summed and the results are stored in a corresponding 32-bit halfword (i.e. half of a 64-bit word); page 6-170]*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify Chehrazi, Oberman, and Peleg to include the sum bytes into halfwords operation and required circuitry as taught by AltiVec.

The suggestion/motivation for doing so would have been that the performance of many common vector algorithms, such as dot-product algorithms, is improved.

13. Referring to claim 11, Chehrazi, Oberman, Peleg, and AltiVec have taught the byte execution unit as recited in claim 8, wherein the pre-processing logic comprises population counter logic *(Peleg; POPCNT Circuits; Fig. 9, components 908a-d)* coupled to receive the operands and configured to produce population output signals indicative of numbers of logic ones in portions of the operands *(Peleg; column 14, line 5-36)*.

14. Referring to claim 12, Chehrazi, Oberman, Peleg, and AltiVec have taught the byte execution unit as recited in claim 8, wherein the pre-processing logic comprises

compressor logic *(Chehrazi; 4:2 compressor; Fig. 4, component 326)* coupled to receive the operands and configured to perform a compression function *(Chehrazi; column 8, lines 10-17)*.

15. Referring to claim 13, Chehrazi, Oberman, Peleg, and AltiVec have taught the byte execution unit as recited in claim 8, wherein the post-processing logic comprises end-around carry logic configured to perform an end-around carry function *(Chehrazi; column 9, lines 45-67)*.

16. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Chehrazi, Oberman, Peleg, and AltiVec in view of "IA-64 Application Developer's Architecture Guide" (Herein referred to as Guide).

17. Referring to claim 14, Chehrazi, Oberman, Peleg, and AltiVec have taught the byte execution unit as recited in claim 8 wherein the byte instruction specifies an average operation *(Chehrazi; column 13, lines 12-20)*.

Chehrazi, Peleg, and AltiVec have not explicitly taught that the post-processing logic is configured to perform bit shift operations.

Guide has taught post-processing logic that performs bit shift operations *[After the addition of the elements is performed (i.e. post-processing) the output is shifted (Guide; "Parallel Average" instruction description on page 7-140 and Fig. 7-29 on page 7-141]*.

At the time the invention was made, it would have been obvious to a person of ordinary skill in the art to modify the post-processing of Chehrazi, Oberman, Peleg, and AltiVec to perform bit shift operations as taught by Guide.

The suggestion/motivation for doing so would have been that performing bit shift operations in post-processing logic allows an average instruction, like that taught by Chehrazi, Peleg, and AltiVec, to execute an averaging operation while preventing cumulative round-off errors (*Guide; "Parallel Average" instruction description on page 7-140*) and without a time-consuming division operation (*Guide; See Fig. 7-29 on page 7-141*).

Response to Arguments

18. Applicants arguments filed on April 19, 2006 and that are still pertinent, have been fully considered but they are not found persuasive.

19. Applicant argues the novelty/rejection of claims 1 and 29 on pages 10, 11, and 13 of the remarks, in substance that:

> "While Cherazi does teach that the multiply circuit is capable of 8x8 multiply operations, Cherazi does not teach the byte execution unit recited in claim 1, for example" (last sentence of third paragraph of section II)
>
> "The mere presence of these three different architectures is evidence that a single byte execution unit for performing the four different byte operations would not have been obvious to a person of ordinary skill in the art" (second paragraph on page 13)

20. These arguments are not found persuasive for the following reasons:

Claim 1 recites the limitation "A byte execution unit". Cherazi has taught a multimedia processor data path (i.e. execution unit) that executes on packed 8-

bit (i.e. byte) data operands (column 7, lines 21-40). The claim language "a byte execution unit" does not require any further teaching. Therefore, Cherazi has taught a byte execution unit as recited in Claim 1.

In response to applicant's argument that the examiner has combined an excessive number of references, reliance on a large number of references in a rejection does not, without more, weigh against the obviousness of the claimed invention. See *In re Gorman*, 933 F.2d 982, 18 USPQ2d 1885 (Fed. Cir. 1991).

21. Applicant argues the novelty/rejection of the limitations of claim 4, now incorporated into claim 1, on pages 11-12 of the remarks, in substance that:

> "Peleg does not provide any suggestion whatsoever that the population count operation of Peleg could be combined with the high performance pipelined data path architecture of Chehrazi" (paragraph at the bottom of page 11)

22. These arguments are not found persuasive for the following reasons:

Peleg has taught a "processor better suited to multimedia applications" (column 4, lines 3-4). One of the ways that the processor of Peleg is suited to multimedia applications is through the use of population count operations on packed byte data elements (column 4, lines 10-19). Chehrazi has taught a multimedia processor that performs operations on packed byte data (column 7, lines 21-40). Since Peleg has taught performing population count operations on packed byte data for multimedia applications, such as those performed by the processor of Chehrazi, and that perfoming such operations improves performances of algorithms that require the totaling of the number of bits set in packed byte data elements (column 4, lines 10-19), there is the

suggestion/motivation for the combination of Peleg and Chehrazi. Therefore, the Office is not using impermissible hindsight for the above combination.

23. Applicant argues the novelty/rejection of the limitations of claim 7, now incorporated into claim 1, on page 12 of the remarks. These arguments are considered moot in view of the new rejection above of claim 1.

24. Applicant argues the novelty/rejection of the limitations of claims 8 and 11-14 on pages 14-16 of the remarks. These arguments are considered moot in view of the new rejections above of claims 8 and 11-14.

25. Applicant argues the novelty/rejection of the limitations of claims 19-25 and 28 on pages 16-21 of the remarks. These arguments are considered moot in view of the indication of allowable subject matter above.

Conclusion

26. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

27. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

28. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

"AltiVec: Bringing Vector Technology to the PowerPC Processor Family" teaches the instructions and microarchitecture of AltiVec Technology and teaches that sum across instructions improve the performance of many common vector algorithms, such as dot-product algorithms *[1st paragraph of section 5 on page 442]*.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin P. Geib whose telephone number is (571) 272-8628. The examiner can normally be reached on Mon-Fri 8:30am-5:00pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Fritz Fleming can be reached on (571) 272-4145. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Benjamin P Geib
Examiner
Art Unit 2181

Fritz M. Fleming
FRITZ FLEMING
Supervisory PRIMARY EXAMINER 6/26/2006
GROUP 2100
AU 2181